

# ASiV 的基础

## TM610



贝加莱工业自动化  
Perfection in Automation  
[www.br-automation.com](http://www.br-automation.com)





## 前提

培训模块:	TM600 – 图文显示的基础
软件:	AS 2.5 或更高版本 / AR 2.80或更高版本
硬件:	4PP220.1043-75 PC 至少256 MB内存 (推荐用512 MB) Windows XP / Windows 2000

1 · 简介	3
1.1 目的	4
2 · 基本信息	5
2.1 什么是可视化组件?	5
2.2 集成的可视化	5
2.3 可视化组件的历史	6
3 · 开始	7
3.1 安装要求	7
3.2 安装可视化组件	8
3.3 附加资源	9
4 · 创建一个新项目	10
4.1 创建项目的准备工作	10
我的第一个可视化组件项目	11
4.2 在第一个可视化组件项目中添加元素	18
5 · 使用可视化组件编辑器	23
5.1 可视化组件帮助	23
5.2 使用可视化组件编辑器	25
5.3 可视化中的组件	26
6 · 可视化组件的实例项目	30
6.1 可视化全局设定	29
6.2 边界和风格	30
6.3 变量管理	33
6.4 页面结构的分层方法	35
6.5 页面分布	46
6.6 虚拟按键和按键动作	50
6.7 静态和动态文本	55
6.8 数字和字母的显示	59
6.9 棒状图及其它图形对象	63
7 · 总结	66

## 1、简介

这个培训模块简单介绍了如何用可视化组件来设计可视化图形，B&R可视化组件正是用来创建简单或复杂可视化应用的工具。

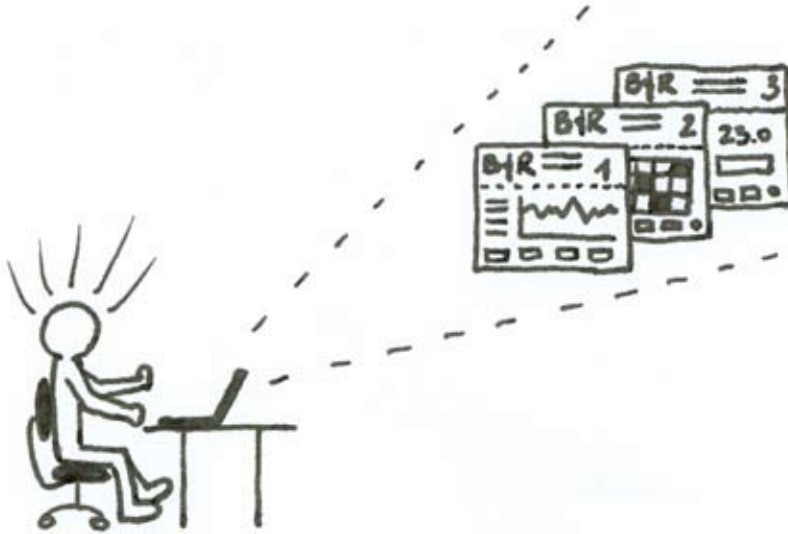


图1 简介 -可视化组件编辑器

通过项目例子的帮助，可以了解在创建可视化应用时哪些步骤是必要的，以及如何有效地使用编辑器组件来创建可视化应用。

## 1.1 目的

掌握如何用可视化组件编辑器创建一个简单的可视化应用。

学习和理解可视化组件的可能性和概念。

培训模块用几分钟演示了怎样创建可视化项目，并在目标系统中运行它。

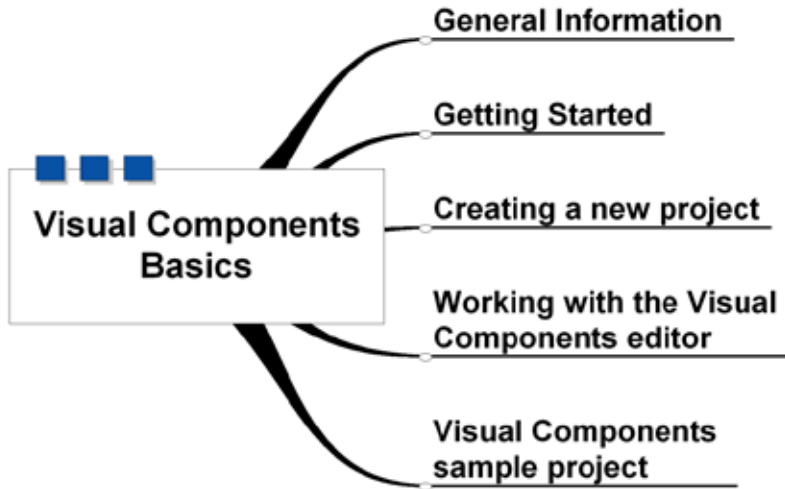


图. 2 棕树

## 2、基本信息

### 2.1 什么是可视化组件？

可视化组件是集成在 Automation Studio 中的可视化。可视化应用（可视化任务）与控制应用程序（控制/定位任务）一起时被管理、编辑和执行。目标硬件不同，显示的范围也不同，可以从2x20线显示到XGA显示(1024 x 768) 最多256色。

### 2.2 集成的可视化

过程图片显示在Automation Runtime Target上。

与一般的可视化不同，运行可视化时会与控制单元分开，显示值通过单独的通讯从控制器中读取—因此可视化会与控制器任务一起运行。

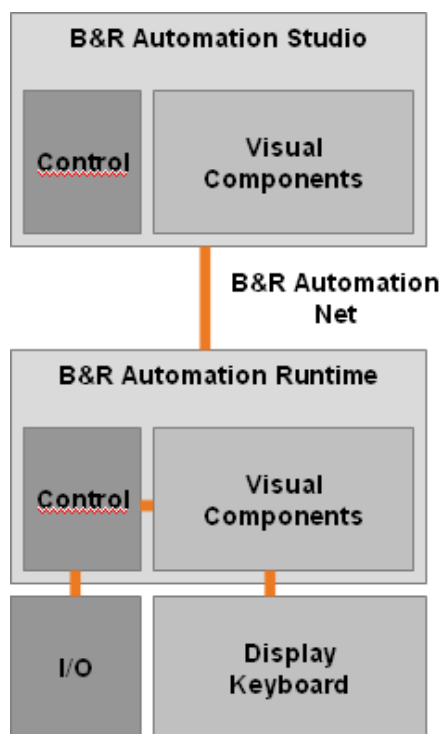


图. 3 集成的可视化

## 2.3 可视化组件的历史

"刚开始时，这里有 ..."

可视化组件的历史背景：

BRRT 操作员面板：

这些面板由软件工程师通过ESC顺序来“编程”。早期，用户必须自己编写可视化程序。后来，为很多面向文字显示和用户输入提供了功能模块。

PCS (PANELWARE 配置软件)— ca. 1992

基于MS-DOS 的 PCS 软件可以实现在PC机上创建和管理可视化。“配置代替编程”第一次变成了现实。项目可以传送给“智能”PANELWARE控制器中，并开始运行。

Panel Studio— ca. 1997

基于Windows 的PCS执行。

可视化组件 (从AS 2.0版本开始) – ca. 2001

集成到AS中的Panel Studio ←→集成的可视化。

SG4的可视化组件 (从AS 2.4.1版本开始) 2004/2005

对于所有的面板，其可视化组件的当前版本都带有VGA显示。现在的AS版本也包含“经典”可视化组件编辑器，可以创建和编辑SG3目标。



### 3、开始

#### 3.1 安装要求

硬件/软件要求:

- PC 至少 256 MB RAM (大型项目推荐使用512 MB 到1 GB的RAM)
- 1 GHz 或更高的处理器
- 操作系统: Windows XP Professional / Windows 2000 Professional
- AS V2.4.1或更高版本

"处理能力与RAM不同, 它与Windows程序的处理速度无关."

运行Automation Studio / Visual Components SG4时需要安装.NET 软件。

如果在 PC上没有安装.NET软件, 可以在 自动化软件工具 CD的目录下 DOTNET\_FW 和所需语言目录(German / English) (dotnetfx.exe)下安装。

开始安装后, 根据选项确认对话框。

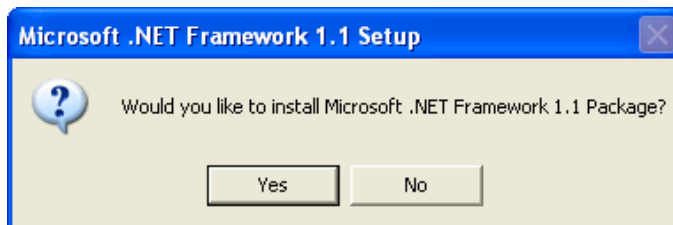


图. 4 DotNet Framework 安装

## 3.2 安装可视化组件

在AS中，可视化组件是一个可选组件。

在安装时，以下组件为必选项 (安装AS V2.5的例子)。

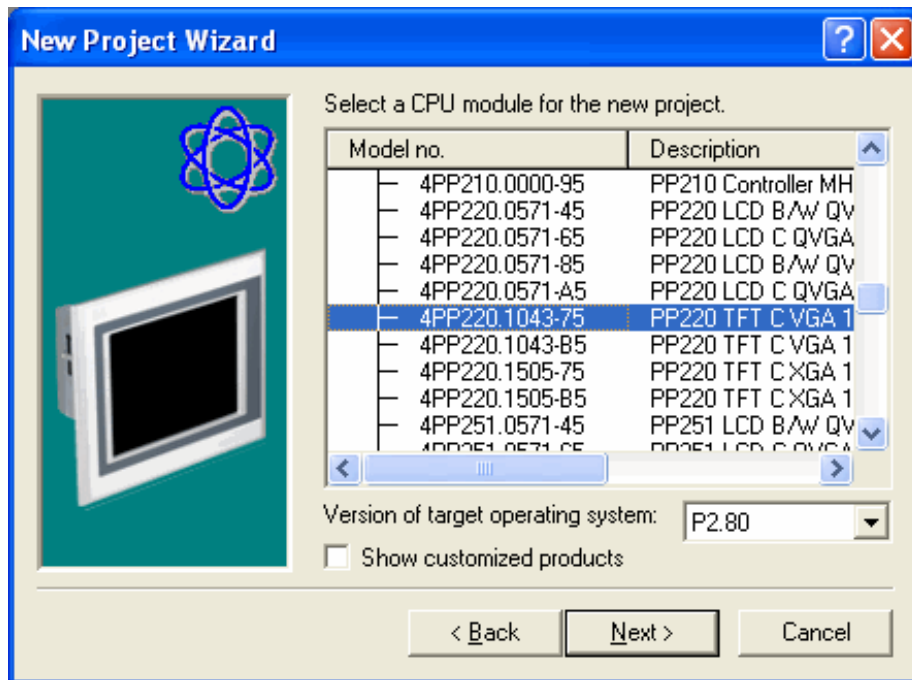


图. 1 Automation Studio安装 – 可选组件

我们也推荐安装"Samples"，因为例子中描述和实现了可视化领域的多种需求。

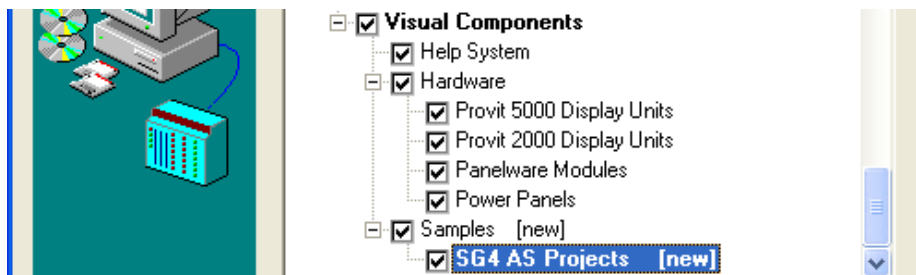


图. 2 Automation Studio 安装 – Visual Components samples

安装完AS后必须重启电脑。

### 3.3 附加资源

"帮助!"

创建可视化组件时，帮助是一个很重要的工具。可视化组件的培训文档中会经常提到帮助文件，因为在创建项目的过程中，帮助文件非常重要并且一直可以使用（只要安装了帮助文件）。



# 创建一个新项目

## 4、创建一个新项目

在创建新项目时，硬件和软件的选择是一个关键部分。

### 4.1 创建项目的准备工作

创建项目之前必须定义硬件。这个培训文档是基于10.4"的PowerPanel(4PP220.1043-75)。



图. 3 4PP220.1043-75

如果这个面板不可用时可使用任何别的SG4目标。



图. 4 5CFCRD.0032-02

传送项目时至少需要24V电压的32 MB CF卡。

建议使用USB口的 CF卡读卡器（台式电脑/笔记本电脑）或PCMCIA / CF 适配器（笔记本电脑），将项目传送到CF卡中（更多信息见下文）。

## 我的第一个可视化组件项目

这一章的目标是创建一个新项目并将它传送到目标硬件中。

可视化组件的所有运行模块都已传送到目标系统中，并且建立了与目标硬件的连接。本质上，通过串口或以太网连接没有关系——但推荐使用以太网连接，可以实现最大传输速度。

### 练习: 创建一个新项目

#### 例子的必要步骤:

- 创建一个新项目，目标硬件为4PP220.1043-75
- PC 和PowerPanel的通讯设置为以太网
- 使用自动化软件工具 "PVI Transfer"将项目传送到CF



## 选择一个新项目并带有目标硬件

当用AS创建新项目时，选择一个PowerPanel（例如，AR目标并带有集成的可视化）作为目标硬件。

# 创建一个新项目

通过菜单 [File] / [New Project] 或 工具栏中 按钮打开"New Project", 首先定义路径和项目名。

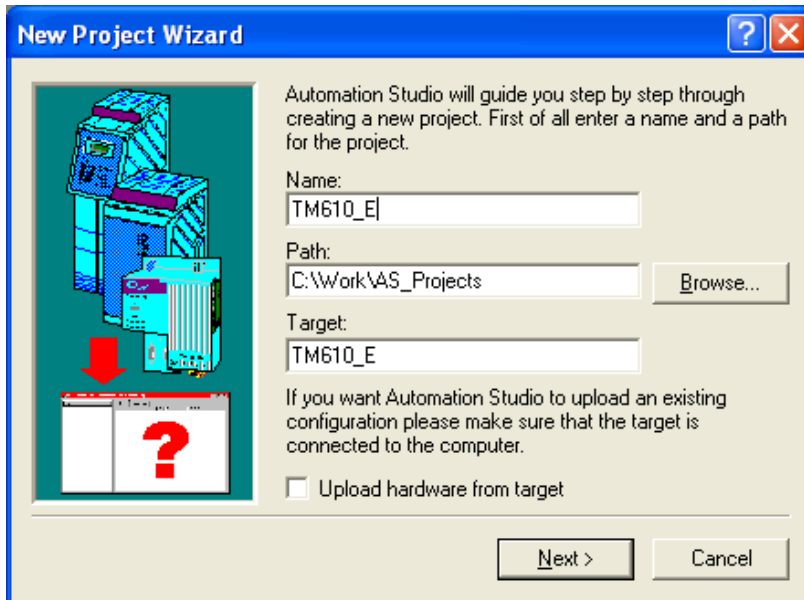


图. 5 新项目

按<Next> 确认对话框中已选中的产品组"PowerPanel".

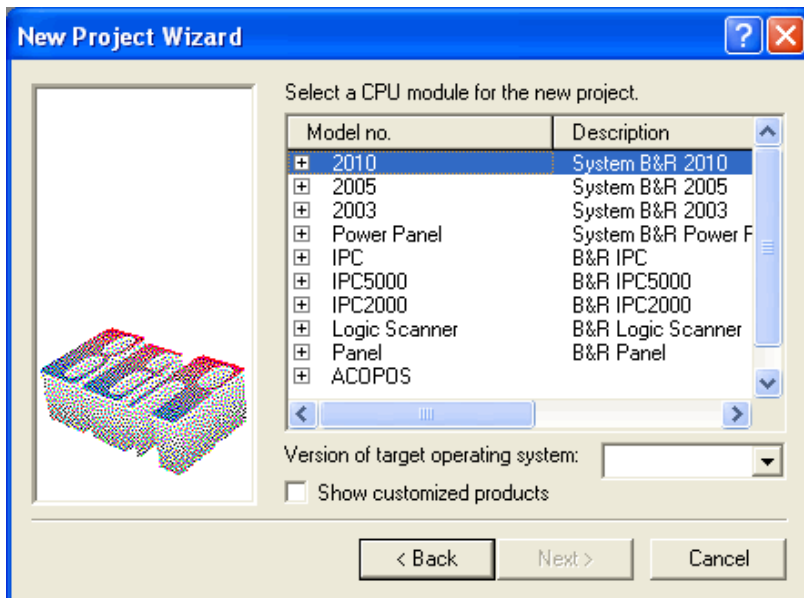


图. 6新项目 – 产品选择

在对话框中选中需要的目标硬件"4PP220.1043-75"。选择选项"Show customized products" 可以显示一种惯用显示。

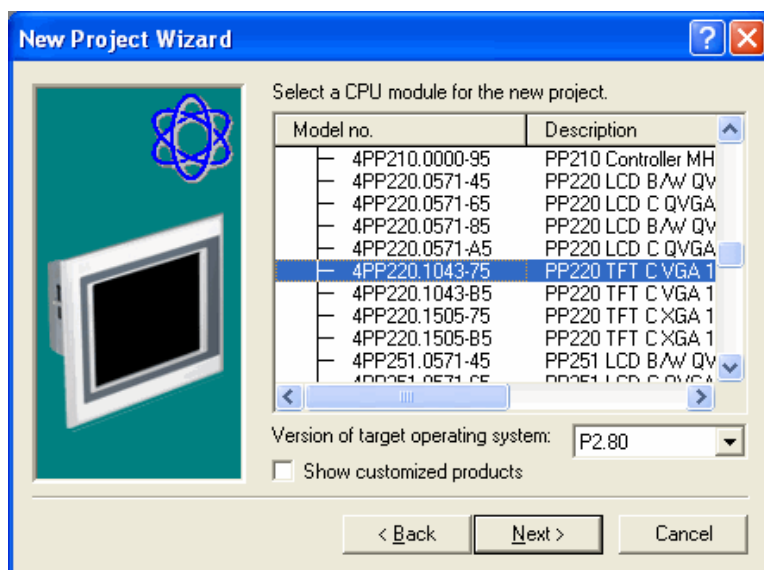


图. 7 新项目 – 组件选择

按 <Next>确认对话框后，在下一个对话框中单击<Finish>关闭项目创建导向。现在，新的AS项目结构已经在硬盘创建好了。

## 创建一个新项目

下一步是定义可视化应用的名字，限制在六个字符以内。在这个培训中，我们定义新建面板的名字为"visu"。

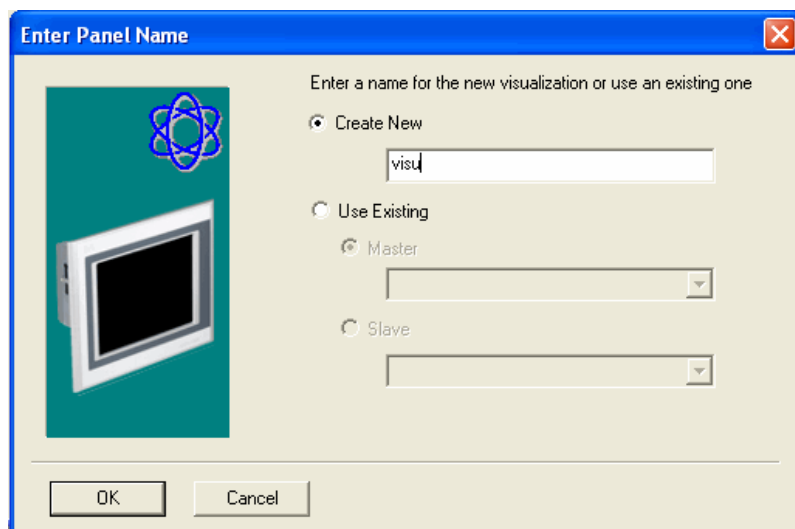


图. 8输入可视化应用的名字

单击 <OK>后，可视化组件工作时需要的所有资源已装载好了，并且可视化组件的编辑器也打开了。

“第五部分：使用可视化组件编辑器”会解答有关这幅视图的所有问题。

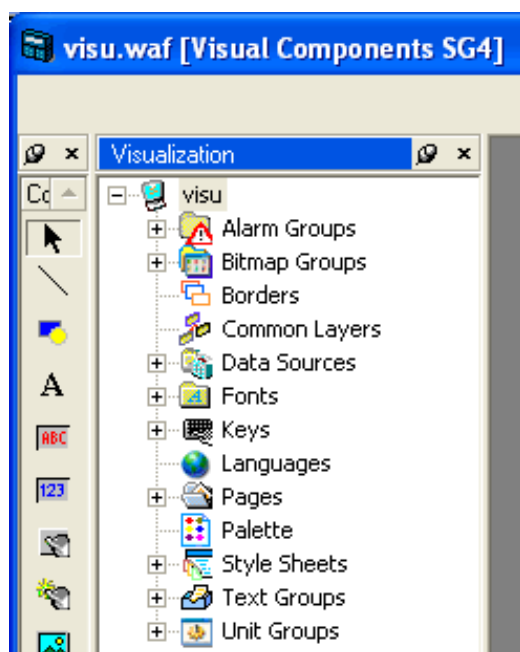


图. 9 可视化组件编辑器 – 项目一览



我们先关闭可视化组件编辑器（即VC编辑器）。

问题

这个练习的目的是什么？

回答

创建一个新项目并将它传送到目标系统中。

关闭VC编辑器后，回到AS项目的画面中。

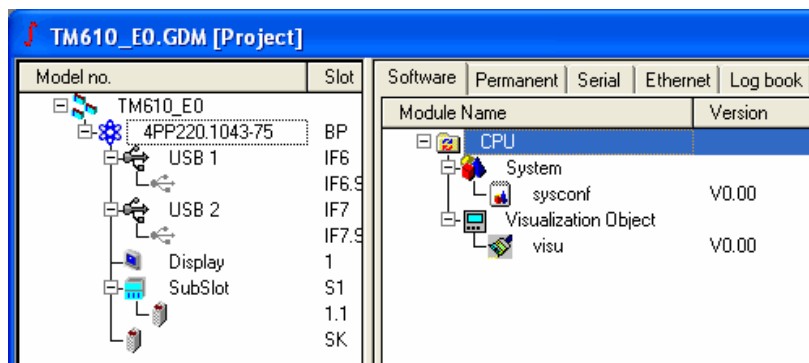


Fig. 10 AS项目画面

## 创建一个新项目

将项目传送到 CF 卡

将项目快速传送到 CF 卡有两种方式：

- AR 操作系统已经安装到CF卡中(项目通过在线连接传送)
- 用自动化软件工具"PVI Transfer"将项目传送到 CF 卡中

使用CF卡时的准备工作

在AS菜单[Tools] / [Generate transfer list]下创建一个 \*.PIL 文件，在PVI transfer tool中使用。

这时生成一个包括操作系统的完整的传送列表。

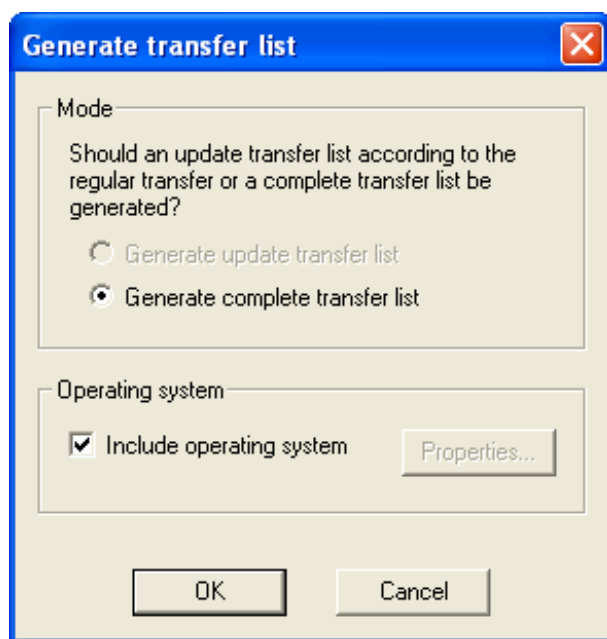


Fig. 11 生成传送列表

点击<OK>创建AS项目，完成传送列表后，在AS输出窗口中会出现"Exit transfer list"。

通过AS菜单[Tools] / [PVI Transfer Tool]会自动打开已创建的传送列表。

在工具栏中点击 按钮打开 "Create CompactFlash"对话框。

在PVI Transfer Tool中接下来几步为:

- 选择磁盘  
在CF卡适配器中选择一个CF卡。
- 生成磁盘  
根据卡的大小，可在一分钟内创建一个可引导的CF卡。

为运行项目准备PowerPanel

- 当PowerPanel 断电时将CF卡插入CF卡的插槽中
- 为PowerPanel提供24V电源。

结果:

- 通电后Power Panel 开始启动，
- 几秒钟之后，启动屏幕变为空白画面。下个练习会添加其它元素。  
这个练习的目标:
- 创建一个新的可视化项目
- 将带有可视化的AS项目传送到CF中
- 测试目标硬件

... 通过练习，花上2分钟左右，就已经启动了Automation Studio。这个时间是从 "New Project"到 "Create CompactFlash"…….

# 创建一个新项目

## 4.2 在第一个可视化组件项目中添加元素

现在在目前空白的屏幕页面中添加元素，第五部介绍如何使用VC编辑器。

练习:在可视化中添加元素

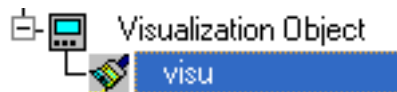
- 定义一个控制变量，
- 打开VC编辑器
- 将输入输出目标连接到控制变量上
- 将变化传送给目标系统

- 创建一个用户任务并定义一个变量

创建一个新的用户任务"panel"，带有本地变量 "ActTemperature"，数据类型为INT。大家已经知道如何实现 (见TM210)。。

- 打开可视化目标"visu"

双击AS软件树中的可视化目标"visu"，打开VC编辑器。



- 画一个输入输出区域，并与变量连接

在目标系统的空白页面上添加一点元素。

打开VC编辑器，用于可视化组件的所有组件会显示在左边。

每个新的可视化组件项目会自动带有几个预先定义好的组件（项目模版）：

- 空白的开始页面 (在上个练习中，目标系统已实现)
- 在触摸屏上，输入数字和字母

这个练习中将执行以下几个步骤：

- 将AS创建的所有控制变量添加到VC编辑器中
- 搜索空白页面
- 插入两个数字区域
- 将其中一个区域的"Input"属性改为"True"，这个属性将这个区域设定为输入区域。
- 用本地控制变量"VisuTemperatur"连接两个区域

将控制任务中定义的变量添加到VC编辑器中

当VC编辑器打开后，并没有显示AS项目已用变量的任何信息。必须在"Data sources"中添加这些变量。

添加变量有两个步骤：

- 选择"Data sources" 项并双击"Local"



- 通过工具栏中的"Update data points"按钮或按<F5>，添加项目中新的数据点。

# 创建一个新项目

在可视化项目中搜索空页

在VC编辑器中，用"pages"来管理过程图像。与Windows的浏览器相似，点击图标 <+> 或 <->可以打开节点的下一层或关闭节点。

双击"InitPage"层，起始页就会显示在屏幕的中心。

双击"Default"层，层以唯一的方式打开，

插入数字区域

只要不隐藏或关闭个人登录窗口，工具栏（即创建页面时所需的全部工具）就会显示在左边。

如果看不到工具栏，通过菜单[View] / [Toolbar]可以打开。



图. 12 基本工具

从工具栏中选择数字控制"123"放到页面中，并点击和拖放矩形到需要的尺寸，重复这个步骤，屏幕显示如下图：

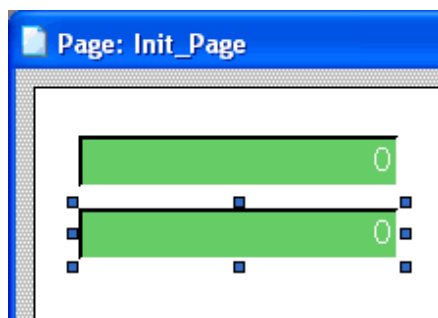


图. 13 插入一个输入和输出领域

在这个练习中，不必将边框、尺寸或设计做得很完美。

从输出到输入区域

如果添加了一个数字区域，那么在运行时作为输出区域来用。

现在将这个区域的属性改为输入领域的功能。

页面中的每个对象都有"属性"，它会影响对象的运行动作。

属性按默认方式显示在右侧。

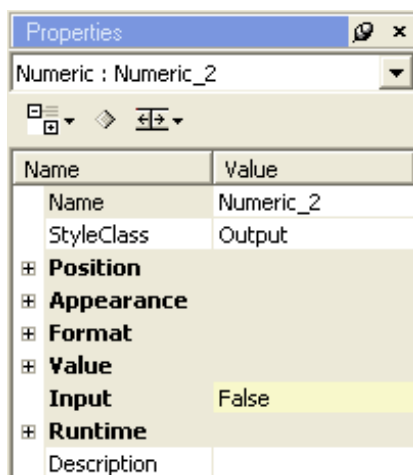


图. 14 输出区域属性

为了将输出区域配置为输入区域，必须将"input"属性改为"true"。这个改变使对象获得一些额外的输入属性，其中一项是用来设置"TouchPad"数字触摸输入的属性。

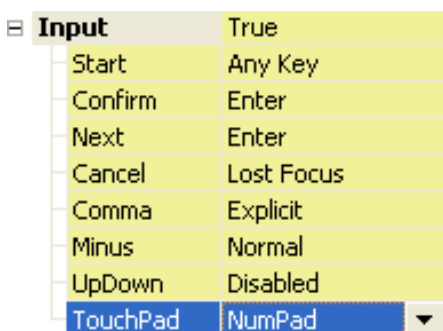


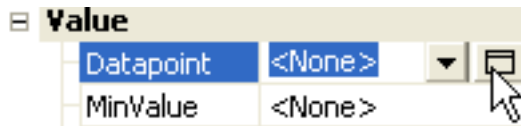
图. 15输入区域属性

# 创建一个新项目

将输入和输出区域与变量连接

如果要输入动态数值或对数值作任意改变，必须将控制变量 "VisuTemperature" 与区域连接，

使用 "Value" – "Data point"属性来连接变量。



控制变量（从现在起作为数据点），通过<Browse>按钮选择需要连接的变量（鼠标所指的位置）。

所有的数据点在新的对话框中显示，并按实用性（局部/全局）来分类。局部数据点在其任务中显示。

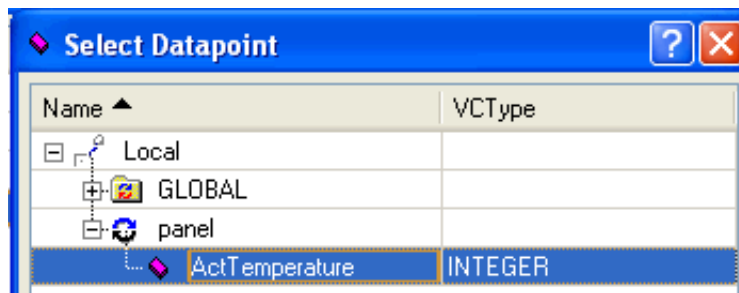


图. 16选择的数据点对话框

局部变量"ActTemperature" 可以在 "panel"任务中找到，选定后点击<OK> 关闭对话框。变量现在已与数字区域连接上了。

第二个数字区域与变量的连接过程相同。

将改动编译后并传送到目标系统前，必须在AS中配置好以太网通讯。

备注:

在AS中配置以太网通讯的信息以及目标系统的信息可以在培训模块 TM212 / TM213中找到。

本次练习目标:

- 通过以太网将可视化项目中的变化传送到目标系统中
- 用一个触摸系统作出你的第一个登录



## 5、使用可视化组件编辑器

现在已经创建了一个可以运行的项目，又知道如何打开可视化组件编辑器（见4.3在第一个可视化组件项目中添加元素），所以我们开始学习有关编辑器窗口和可视化目标的内容：

- 使用可视化组件帮助来工作
- 使用VC编辑器
- 使用VC编辑器中的组件

### 5.1 可视化组件帮助

当使用VC编辑器来寻找相关功能和组件的详细描述时，可视化组件的帮助（从现在起记作VC Help）是一个很重要的辅助工具。

这个培训模块偶尔也会提到VC Help来回答问题以及学习如何使用这个帮助。

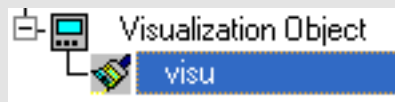
练习：可视化组件编辑器窗口

打开帮助窗口，找到“Visual Components Editor window”的基本描述。

方法：

在VC编辑器中，按<F1>键打开VC Help，

双击可视化目标打开VC编辑器。



结果

找到菜单选项 “Finding your way in the editor”

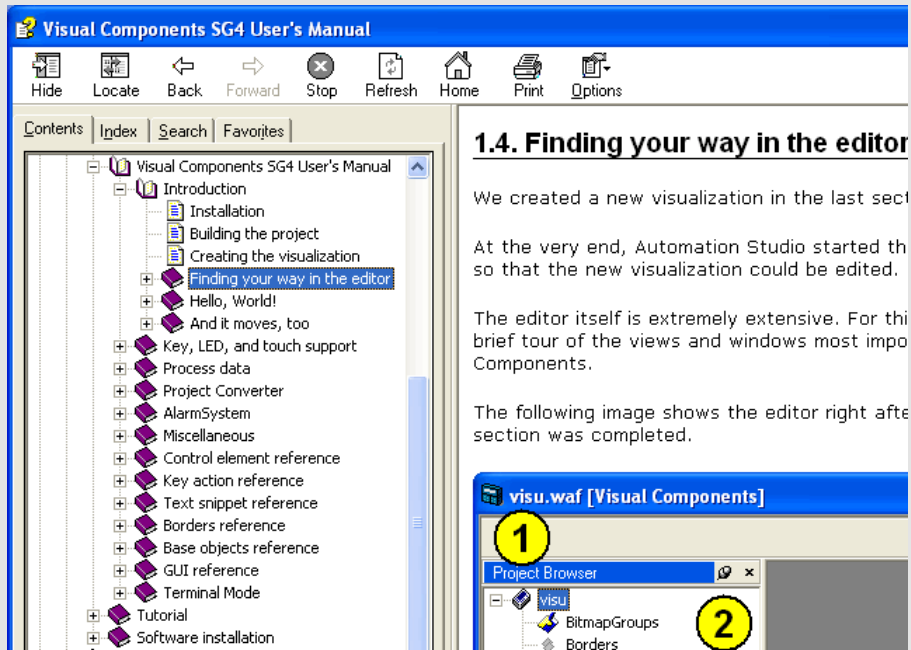


图. 17 可视化帮助 – Finding your way in the editor

提示:

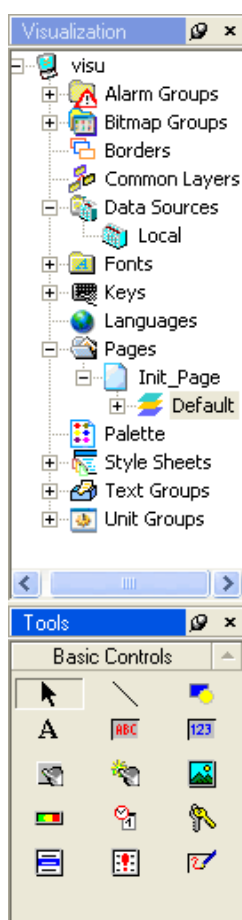
打开帮助后，使用<ALT> + <TAB>组合键在AS和帮助之间切换。

## 5.2 使用可视化组件编辑器

这部分介绍如何使用可视化组件编辑器。已经使用过可视化工作的人会感到好像在家里使用程序一样。

用户可以编辑VC编辑器中的每个窗口，即可以隐藏窗口或将窗口移动到希望的位置。

在默认情况下，可视化树和主控制在屏幕的左边显示。点击"Pin" 可以隐藏一个窗口。双击标题栏，可以从框架中释放一个窗口并把它单独放到屏幕上。



练习：改变编辑器窗口的设计

接下来的几分钟，试着改变编辑器窗口的设计。

这个练习的目的

理解编辑器的对接行为。

图. 18 可截短的窗口

## 5.3 可视化中的组件

默认的项目

当创建了一个新的可视化后，就可以立即编译并运行它（正如第一个练习中所见到的）。

默认的项目包含：

- 系统警报(TM640)
- 触摸板和报警系统的位图组(TM640)
- 单位组(TM650)
- 边界
- 默认字体(Bitstream Vera Sans)
- 起始页面(Init\_Page)
- 德语和英语 (TM650)
- 彩色和单色的风格页
- 触摸板的按键设置
- 报警系统的文本组 (TM640)

备注：

这个模块不介绍报警系统和多语种方面的内容(TM640, TM650)。

第一个练习中，有个名为"Init\_Page"的空白页会显示在目标系统中。

练习：打开各个编辑器

在这个练习中，我们将打开可视化组件项目树中的每个编辑器，

这个练习的目的

熟悉可视化组件项目中的组件，下一部分是使用这些组件来工作。

通过这个练习，可以在屏幕上打开几个编辑器。

如果打开的窗口过多，项目会变得有点不清晰。通过菜单[Window] / [Window]或组合键<CTRL> + <ALT> + <W>打开窗口对话框。

对话框可以关闭一个或所有已打开的窗口。

## 备注:

每个打开的窗口都需要占用Windows内存。如果窗口打开太多，程序就会变慢。当退出编辑器时保存当前画面，当重新打开编辑器时会恢复当前画面。因此，项目打开的时间会长一点。

## 可视化组件中的常规

当使用可视化组件编辑器时也应该遵从几个指导方针。

### 任务: 可视化组件的常规

从可视化组件帮助中了解并学习可视化组件的常规，常规为什么重要？有些什么样的常规？

### 任务帮助

常规位于 "Miscellaneous"下。

## 6、可视化组件的实例项目

在前面的练习中，我们已经学到一些关于VC编辑器及其组件的内容。

在这一部分，创建一个名为“real”的实例项目并运行它，用来解释VC中的主要组件。

在每个项目的步骤中会执行不同的任务和练习，如下图所示的实例项目。

"通过实践来学习!"

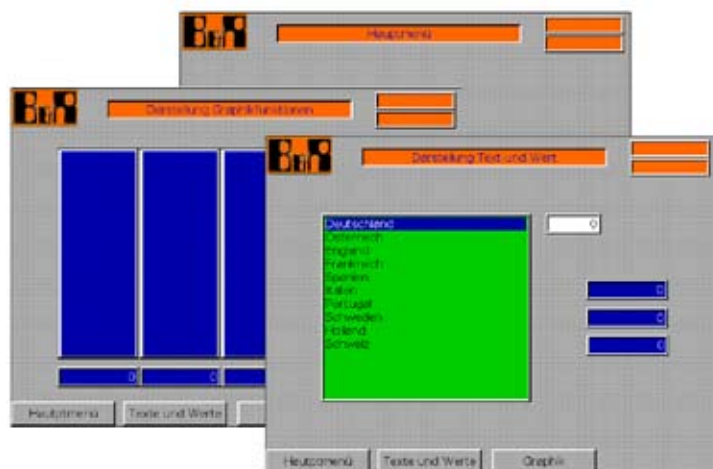


图. 19可视组件编辑器 – 实例项目

解释和执行以下的项目部分：

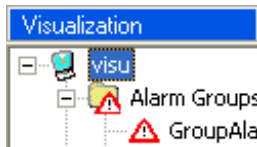
- 可视化全局设定
- 边界和风格页
- 变量管理
- 页面结构中的分层方法
- 页面分布
- 虚拟按键和按键动作
- 静态和动态文本
- 数字和字符显示
- 棒状图和图形对象

## 6.1 可视化全局设定

可视化应用具有全局属性，它可以影响可视化的运行行为：

- 默认语言
- 起始页和风格页面的彩色/单色设定
- 屏保（时间定义和页面选择）
- 运行行为（例：通过变量改变页面，通过变量切换语言等）

在VC编辑器中，可视化树的最高层可以设定属性。



在页面属性中进行配置。

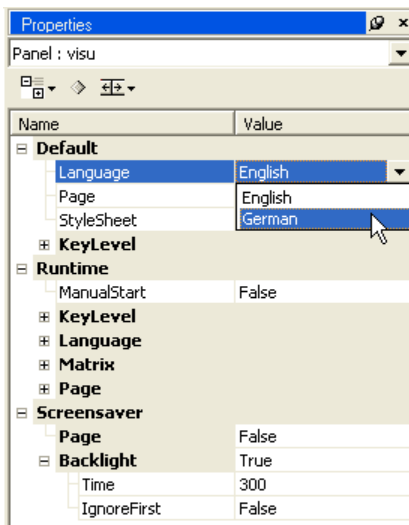


图. 20 全局项目属性

任务: 定义默认语言

选择语言"English".

所有文本以这种语言在目标系统中显示，但可以通过用户任务中的变量来改变它。

## 6.2 边界和风格

### 6.2.1 边界

几乎每一个可视化对象，如文本域，数字域或按钮都有边界。这些对象的外观不是静态的，可由用户来设计。

在默认项目中，已提供了大多数视图的边界。现在为显示静态文本添加不可见的边界。

但不可见的边界有什么优点呢？

在可视组件中，边界可以边界本身（风格）决定，也可以由"Padding"决定，它定义了文本或图形（内容）到边界边缘的空间。

双击可视化节点"Borders"会显示项目中所有的边界。

练习：静态文本的边界

创建一个不带边沿的新边界，显示静态文本。



方法

名为 "Border\_1" 的 "Invisible" 边界可以通过 <Insert> 键或文本菜单 [Add border] 来插入。

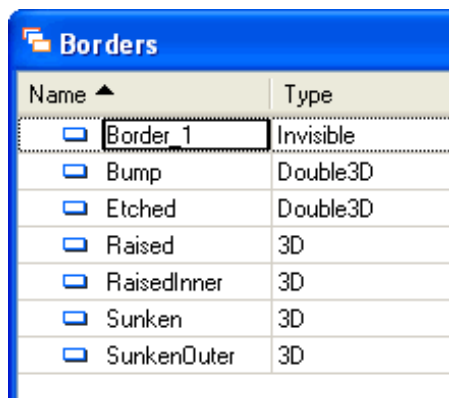


图. 21 组件 – 边界

在 "Name" 属性中命名这个边界为 "Static\_Text"。类型 "Invisible" 定义了边界是不可见的，且缩进尺寸为水平2像素，垂直1像素。



## 6.2.2 风格页

对象的外观（也称为模板）由风格页来定义。正如第二个实例中所见（关于输入输出区域），这些区域已有一些预先设定的属性（如前景和背景颜色）。

风格页中的选项可以设定对象的外观，这样就不用再在创建页面后再调节它的属性了。

风格页的优点：

- 属性的集中管理  
例：输入输出区域的颜色设定
- 控制的多种风格类别  
例：温度，过程指示的多种棒状图
- 颜色设定的中心管理显示彩色或单色
- 整个项目在一个位置改变属性

任务: 改变输入区域的背景颜色



这个练习将介绍如何改变输入区域的背景颜色。

在输入区域的属性中可以直接改变，如果再添加一个输入区域，其颜色属性需要再次修改。

更改后，可以在目标系统中检查结果。

方法

在风格页中，背景颜色以数字方式来控制（PP220为彩色显示）

在属性组里, 由于控制的外观改变了，所以"Appearance" 一定也改变了。

## 6.2.3 设定值和实际值的风格页

任务: 输入和输出区域的风格

这个任务可以分别执行:

- 为显示实际值（白底黑字）创建一个新的风格
- 为输入设定值（蓝底白字）创建一个新的风格



方式

在风格页中，为控制 "Numeric" 添加两个新的风格类型 "OutputActualValue" 和 "InputSetValue"，其外观和输入动作都在属性中定义了

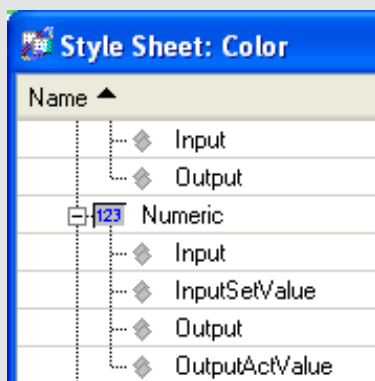


Fig. 22 风格页 – 输入和输出

## 6.3 变量管理

可视化应用显示了处理过程的顺序并允许操作员干涉处理过程。过程通过可视对象如棒状图或数值来显示，在实际操作中使用屏幕上的按键或按钮。

在可视组件中，用户任务中的变量是作为数据点来管理的，同时它也是过程顺序（控制应用）和可视化之间的接口。

### 6.3.1 局部数据源

在第二个练习中，已经在可视化中插入了一个数据点。



控制任务中的变量作为“局部数据源”中“数据点”导入到VC编辑器中。

如何显示可视化任务"panel"中的变量 "ActTemperature"?

Name ▲	PLCTy...	VCType	Unit Group / Subtype	Limit	PLCUnit
GLOBAL					
panel					
ActTemperature	INT	INTEGER	None	None	None

Fig. 23 数据源 – 局部

变量会直接显示在相对应的任务中，因为变量定义为"local"。

如果在控制任务中改变或添加了变量，那么按<F5>或工具栏中的 按钮随时更新视图。

#### PLC 类型

在控制项目中，用户任务定义的数据类型会显示在PLC类型下面。

# 可视化组件的实例项目

## VC 类型

VC类型定义了数据类型，这些数据类型在可视化中使用。VC类型取决于控制变量的数据类型，包含的数据类型有：

- <None>  
这个设定表明在VC中不能将变量作为数据点来使用。
- CHAR  
单个字符，但输出动作由控制器的数据类型来定义（8或16位）（UNICODE = 培训模块TM650）。
- INTEGER  
一个完整的数值，其范围取决于控制变量的数据类型（最大INT32）。
- SCALED  
当缩放变量时，应选择这个数据类型（培训模块TM650），实型变量可以直接作为SCALED显示。
- STRING  
过程变量区域（数组）共同组成一个字符串（也可见CHAR）。

在这个培训模块中，使用VC 数据类型INTEGER、CHAR和STRING来工作。

## 更多信息：

- 可视化组件帮助---处理数据
- 培训模块 TM650

### 备注：

有关UnitGroup, limitation 和 PLC 单位的详细内容可在培训模块 TM650 中看到。

## 6.4 页面结构的分层方法

分层方法是指在一个页面上可以使用多层（例：页眉层、主页面层、页脚层等）。

分层方法的优点：

- 在同一位置经常显示的图像信息只需画一次。
- 在一个位置就可以改变全局图像的信息。
- 当编辑和创建图像时，可以隐藏层或以框架形式简单地显示对象。

因此，全局（公共）层和局部层之间有些差别。  
关于局部层和公共层的数目或混合数目没有限制。

局部层

局部层只能在创建它的页面上使用

公共层

全局层可以在多个页面上重复使用。即在多个页面上使用的图像信息只需画一次。

### 练习:页眉和页脚的普通层



现在创建两个公共层来显示页眉和页脚。同时，也将了解那些已添加到这个层里的对象。

选择 "Common Layers"并点击<Insert>键，可以插入一个新的空白层并显示在屏幕中心。

这个新层叫做"Header" ("Name"属性)。

现在重复这个过程，添加一个新层"Footer"。

节点视图显示出有两个新的公共层。

### 使用哪种语言？

在全局项目的设置里，已经决定了在运行过程中使用英语（默认状况，项目创建后包含两种语言）。

在一个层中，必须决定VC编辑器的文本输入使用哪种语言。从工具栏的组合框里选择"active language" (=创建项目的语言)。

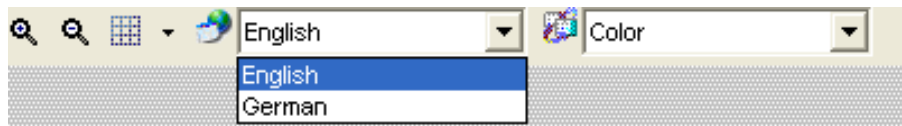


Fig. 24 工具栏 – 语言选择

在页眉中显示:

- 标志
- 日期和时间
- 页面描述

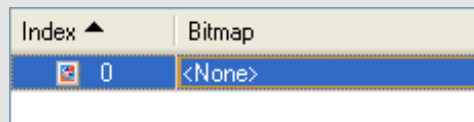
## 6.4.1 显示一个位图

从显示位图开始（例：页眉中的公司标志）

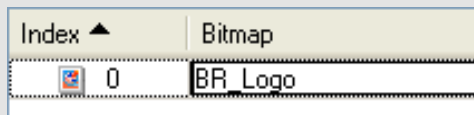
练习：在位图列表中加入一个位图



- 使用 <Insert> 键在"Bitmap"节点中添加一个新组"Logos"。
- 将位图插入新位图组。点击编辑器区域的屏幕中心，空白处。
- 按<Insert>键，一个新的位图元素就会添加到屏幕中心，但还需要连接。



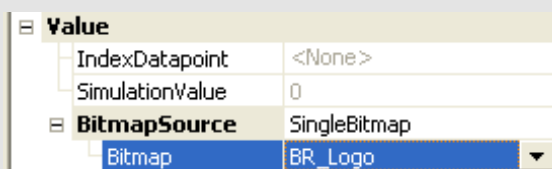
- 位图还没有分配"BitmapReference"属性，从 <None>就可以看出。
- 使用 <Browse>按钮 打开"Add bitmap"对话框。
- 可以在硬盘上搜索我们的标志，点击<Browse>按钮将位图插入位图列表。





练习：在页头中显示标志

- 双击打开“header”层，在编辑器窗口中会显示一个空白页。
- 在页面的左上角放置一个位图控制，尺寸并不重要。
- 位图在"Value"属性中进行连接。这个属性还有另一层"Bitmap Source"，在这里用户可以选择使用单个位图还是多个位图。连接变量后，通过控制程序中的数据点可以动态切换位图。
- 在 "Bitmap Source"属性中，从"Bitmap"属性下的组合框中选择我们的标志。



现在位图以适当的尺寸显示在页面中，所有放在页面中的控制会显示在“header”层中。



使用"Name"属性将位图命名为"Logo"，因此以后就可以通过名字识别标志了。



## 6.4.2 显示日期和时间

日期和时间分别显示在页眉（例：在两行显示）。

### 练习：日期时间控制

显示日期和时间要有以下几步：

- 在页面的右上角放置两次"DateTime"控制。
- 改变两个控制的外观（例：“Appearance”属性），将"Border"设置为"Bump"。
- 改变两个控制的"BackColor"属性为桔黄色。
- 改变"Format"属性；"Alignment"为"Horizontal", "Vertical"为"Center"（例：中心文本显示）
- 改变两个控制的"Name"属性为"Date"和"Time"。



### 问题

为什么我们在属性中看到不同的黄色背景阴影？

### 答案

风格页中的属性以浅黄色显示，而改变了的属性以深黄色显示。

另一种可能是改变了控制风格页，即只需在一个地方做改变。

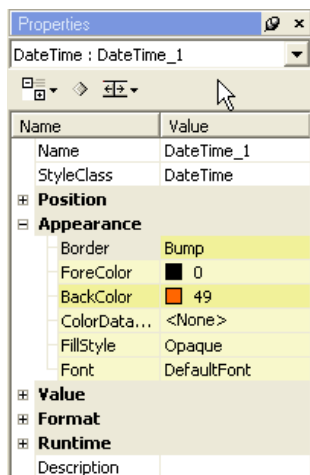


Fig. 25 属性 – 日期时间

现在我们已经调整好如何显示控制了，下一步就是分配数值的地址（如"Value"的属性），第一次控制分配给date，第二次控制分配给time。

## 可视化组件的实例项目

设定日期和时间的格式

当设定日期和时间时，控制任务中的变量(如：读取时间并复制给结构)不会象想象的那样显示。

这是由于运行时的显示限制。例：不可能显示这种语言形式(DD.MM.YY – MM/DD/YY)。

现在有两种方式显示日期和时间：

- 将控制变量作为数据点分配给"Value"属性—秒数从1.1.1970开始。如果属性没有连接数据点，那么就使用内部时钟。
- 设定日期/时间格式，用户可以自由定义"Format"属性—"Source = Single Text"—"Text"。

我们仅使用第二种选择，它可以设置为各种语言。

问题

在"TextGroup"属性中的"(embedded)"值是什么意思？

答案

无论是“text group”（一组中管理多个文本输入）还是"embedded"文本，(如：直接文本输入)都可以输入这个属性。

我们的例子使用"embedded"文本。即我们可以在文本属性中将日期和时间的语言格式指定为"English"。

时间格式：时：分：秒

%H:%M:%S -用大写字母定义时间格式

日期格式：日.月.年

%d.%m.%y -用小写字母定义日期格式

## 6.4.3 静态文本显示

页面描述(页面名称)应该显示在文本区域中。

显示文本的可能方式:

- 嵌入式文本 (如: 在文本控制中直接定义文本) .
- 在文本组中定义文本。

我们在此也只使用第二种选择, 因为这个在开发中有一点优势。

- 静态和动态文本的中心管理
- 在其它位置可以重复使用文本

### 练习:创建一个文本组并插入文本



在文本组中管理页面描述的文本。由于例子只有一页, 所以文本组从文本 "Main menu"开始。

点击文本组节点中的<Insert>按钮, 创建一个新的文本组。

在编辑窗口中显示编辑对话框。项目中使用的每种语言, 都可以在此插入不同的文本线。

现在将文本组 "TextGroup\_1" 重新命名为 "PageDescription", 以便以后快速查找。

最后点击<Insert>按钮, 在编辑器窗口中添加新文本。文本 "MainMenu" 既可以输入到相关语言的编辑窗口下, 也可以输入到 "Text"属性下。

### 备注:

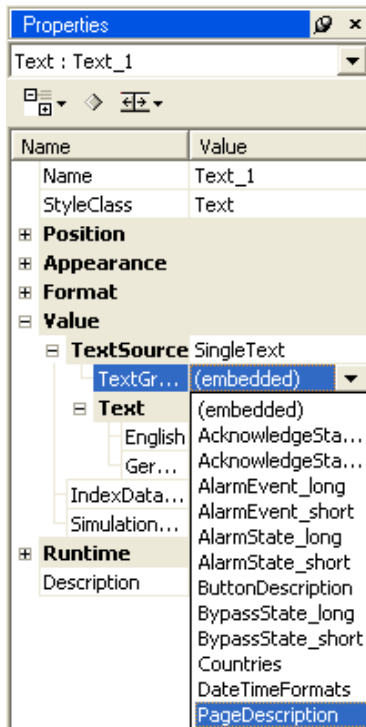
推荐在属性列表中作改变, 因为所有的属性都罗列在此。而文本组是一个例外, 因为它必须使用编辑窗口。

### 练习: 画一个文本框并指定文本



将视图切换到公共层 "header", 在页眉的中心, 通过拖拽一个边界输入一个文本控制。

# 可视化组件的实例项目



在文本控制的文本属性中，将文本组中的一个文本指定为"Embedded"文本。首先，选择文本组“PageDescription”作为显示，如右图所示。

索引为"0"的文本为默认显示文本。如果文本组包含多个文本，那么需要从复合框中选择所需要的文本。

Fig. 26 文本控制中的文本组

任务:改变文本控制属性

改变文本的外观，学习如何使用属性来工作。

方法

例如，改变如下所示：

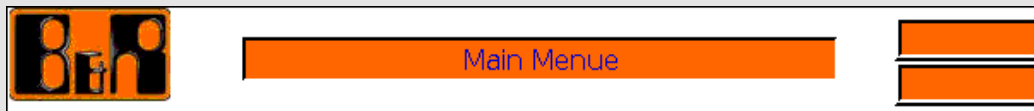


Fig. 27 Example – 公共层页眉

页脚应显示什么？

- 切换页面的按钮

用户使用页脚中的按钮来切换页面。当结束培训时，培训例子包含三个页面。因此，我们需要两个新的文本输入（最好在一个文本组中管理）。

第一步，画出并设计按钮。定义要执行的动作（页面切换）等。

## 6.4.4 按钮和热点

有两种方式可以实现一个按钮：

- 可见的按钮 = 按钮
- 不可见按钮 = 热点

按钮 ↔ 热点的差别

这两种按钮的功能(如：执行的动作)是一样的。差别在于可见的按钮也有可视的组件，如边界、颜色以及显示文本或图形的选择。

热点，是需要静态或动态对象的图像区域中执行动作时来使用。

练习:在页脚中画一个按钮



在公共层"footer"输入三个大小相同的按钮并且相邻排放。通过拷贝和粘贴或使用工具栏中的Alignment icon"来实现。

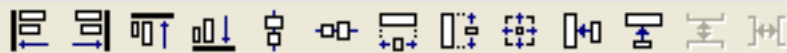


Fig. 28 工具栏 – 排列控制

任务:对齐按钮

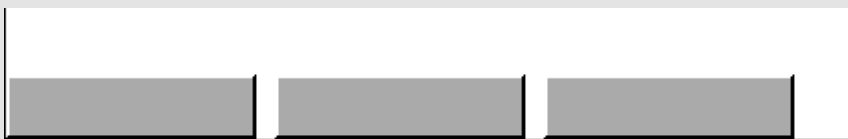


当分别插入三个按钮控制后，使用"Alignment"工具将每个按钮做成相同的大小。另外，给每个按钮起有意义的名字，因为Button\_1，Button\_2，Button\_3不具有多少解释性。

颜色可以通过改变按钮属性来选择。

方法

插入一个新风格，按钮的设计可做全局性的改变，页脚看起来如下图：



# 可视化组件的实例项目

在按钮中显示文本

创建一个文本组，带有按钮描述并输入以下文本：

- 主菜单
- 文本和数值
- 图形

备注：

在按钮中也可以使用页面描述中的相同文本。但页脚中的按钮描述通常比页眉中的图像描述短。

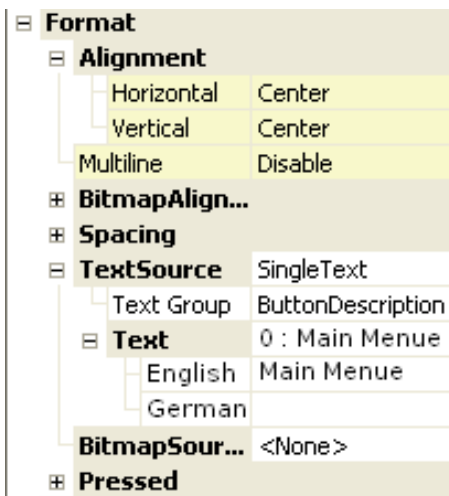
按"Text property"的顺序将文本分配给每个按钮。

## 练习: 指派文本到按钮控制



创建一个名为"ButtonDescription"的文本组以及按钮描述的文本，这些文本将会分配给每个按钮。

将文本分配给按钮控制会比文本控制复杂一点



按钮的外观不是静态的。按钮检测“released”释放和“pressed”按压状态。因此，必须配置每个状态的外观。

如果按压和释放状态的文本相同，那么将“Pressed”属性中的“TextSource”设为<SameAs>。

右边的图像显示如何根据要求来配置按钮。

Fig. 29 按钮属性 – 分配文本

## 练习: 指派文本到按钮控制

创建一个名为"ButtonDescription"的文本组以及按钮描述的文本，这些文本将会分配给每个按钮。

将文本分配给按钮控制会比文本控制复杂一点

页脚应显示的按钮如下:

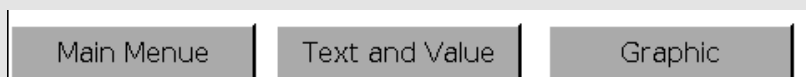


Fig. 30 页脚 – 切换页面

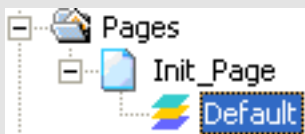
## 6.5 页面分布

这一部分学习如何在例子中创建所需的页面，并集成了页眉和页脚。我们可以在目标系统中看到结果。

练习:在起始页中添加全局层



首先选择起始页，



起始页名为 "Init\_Page"，也包含一个名为"Default"的层。

第二个例子中，在这个层添加数字输入和输出区域。

在起始页"Init\_Page"的"Context menu"中，将一些参考量添加到这些层中（“header”和“footer”），这样做是为了给起始页粘贴一个全局（如：公共）层。

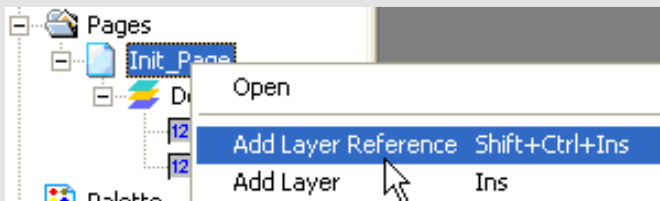
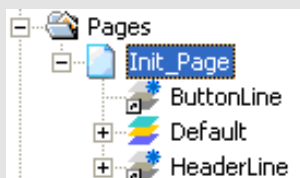


Fig. 31 添加层参考量

公共层会显示在"起始页Init\_Page"的“Default”层下。





在编辑窗口中，页面显示如下：

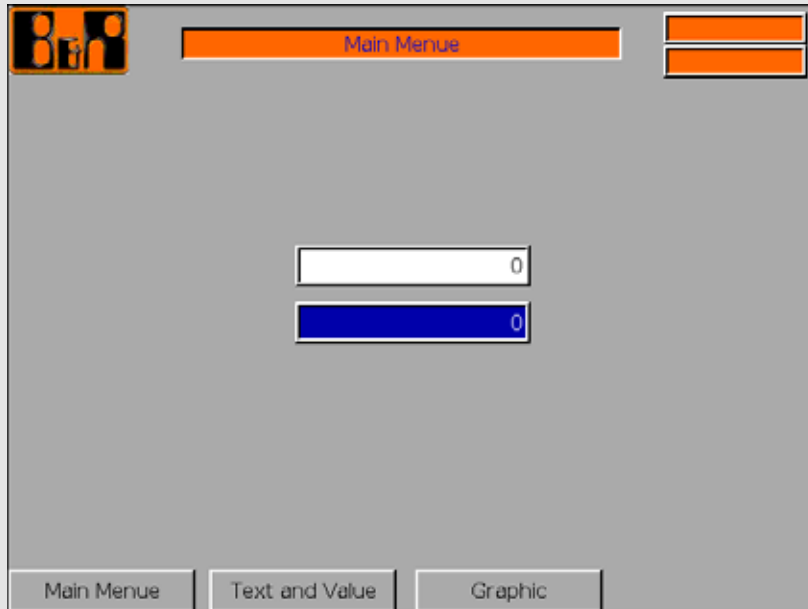


图. 32 示例项目 – Main menu

任务:将改变的内容编译并传送到目标系统



起始页在目标系统中输出，但按钮还没有功能，因为没有创建其它页面，也没有对按钮指定任何功能。

在两个DateTime控制中都显示了时间。

## 问题

为什么定义过的前景和背景颜色不能显示在目标系统的输入区域？

## 答案

输入区域也有焦点颜色。如果在起始页中只有一个输入区域，那么它会立刻成为焦点并显示焦点颜色。焦点颜色可以在页面属性中设置，也可以被取消。

# 可视化组件的实例项目

## 6.5.1 页面属性

每个页面都有具体的属性。最重要的属性是那些在运行时会影响页面外观的属性，如背景颜色和全局输入动作（例：此页上的焦点颜色）。

VC帮助详细描述了这些属性。

## 6.5.2 创建新页面

下一个任务是创建另两个页面，“Text and Value”和“Graphics”。

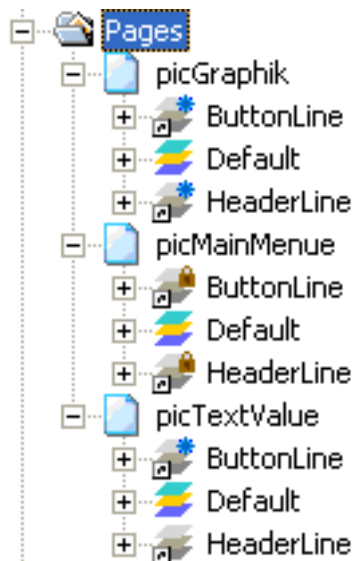
从“Pages”节点开始，点击<Insert>按钮添加一个新页面。

任务：创建一个新页面并插入全局层

创建页面并命名为相应的名字，在每个页面插入“header”和“footer”。

结果

现在项目树的“Pages”会显示如下。



如项目树中所示，页面“Init\_Page”重新命名为“picMainMenu”。

## 6.5.3 管理页面中的层

一个页面包含一个“Default”缺省层或多个层，这取决于页面的功能。

这些层在VC编辑器中是分别管理的，（如：可以防止在编辑的过程中转移或删除其它层上的对象）。

此外，可以使用工具栏将每一层锁定，用来编辑、隐藏或显示。

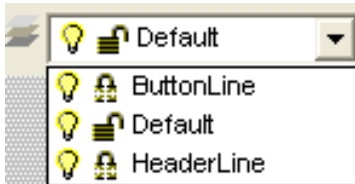


图. 33 工具栏 – Layers

当打开ComboBox时，会显示页面中使用的所有层。

可以实现以下功能：

- 使用灯泡的图标隐藏一个层上的所有对象
- 锁定，释放或冻结一个层上的所有对象

任务：改变层的可见性/不可见性



现在了解一下在任何页面中，切换机械装置的可见性与不可见性所带来的各种影响。

## 6.6 虚拟按键和按键动作

到现在为止，已经创建了几个页面并添加了切换页面的按钮。这一部分主要描述“virtual keys”虚拟按键和“key action”按键动作的功能。

### 6.6.1 什么是“虚拟按键”？

"按键动作的设置，不依赖于硬件"

“虚拟按键”与物理按键或按钮无关，它描述的是使用按键动作的按键行为，而动作发生的位置并不重要。

也可以拿全局层来做比较。位于这层的图形对象，本身并没有功能。这些层由于链接到页面上而“获得生命”。

现在将这个功能也应用于虚拟按键。虚拟按键只有连接到按键或按钮时，它所分配的动作才能执行。

所有的“虚拟按键”可以应用于整个可视化项目中。

这样就可以给虚拟按键只分配一次按键动作，接着能以按键（按键显示）或按钮（触摸屏显示）的形式放置在多个位置。

练习：创建一个“虚拟按键”来切换页面

双击“Keys”节点，在编辑器窗口中打开“Virtual keys overview”。



从现在起，以VK代表Virtual keys。

默认项目中提供的所有 VKs 都显示在"Virtual key overview"中。这些 VKs 用于所提供的TOUCH输入对话框中。

使用组合键<SHIFT> + <CTRL> + <Insert>或在总览的快捷菜单中添加一个新VK。

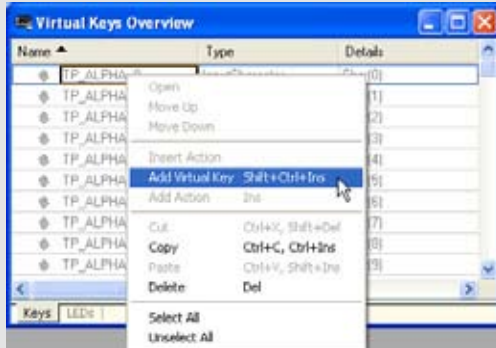


图. 35 虚拟按键总览

现在添加一个新的VK。

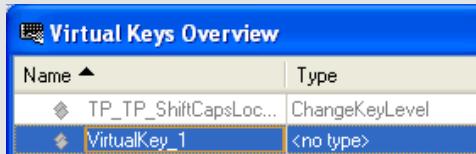


图. 35 新的虚拟按键

在属性中设置这个VK。

给这个VK分配一个默认的名字和动作。

## 可视化组件的实例项目

在动作中定义一个页面的更改。更多的属性是根据定义的动作来显示的。  
在我们的例子中，必须指明页面更改的目标。

任务：创建一个改变页面的VK

在示例项目中，页面应创建三个带有"ChangePage"功能的VKs。



结果

◆ VK_picGraphic	ChangePage	Target(Page, picGraphik)
◆ VK_picMainMenu	ChangePage	Target(Page, picMainMenu)
◆ VK_picTextValue	ChangePage	Target(Page, picTextValue)

任务：按键动作

我们要熟悉一下各种按键的动作，并花时间查看一下这些动作引出的属性。



注意：

按键动作及其属性的详细描述可以在VC帮助中找到。

## 6.6.2 将VK分配给按钮

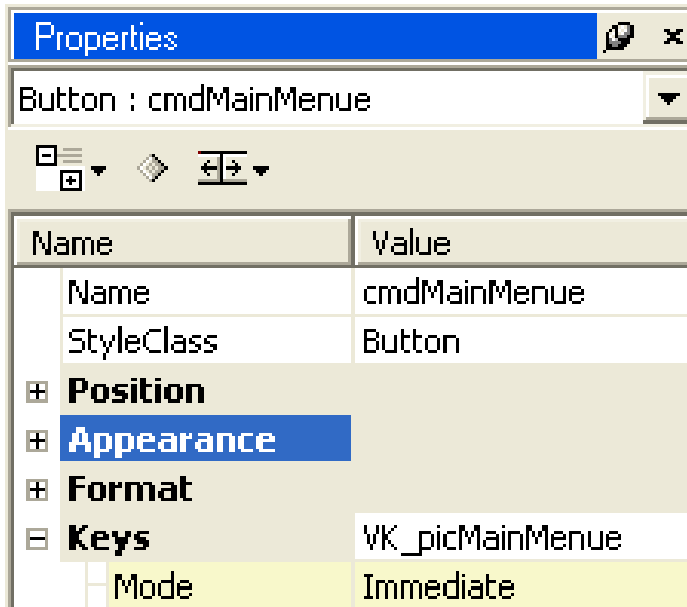


图. 36 将“虚拟按键”与按钮连接

现在将新的VKs分配给"footer"层的各个按钮，这样才能执行"ChangePage"的动作。将VK分配给"Keys"属性。

## 6.6.3 直接在控制中创建虚拟按键

可以直接在按钮、热点或键的"Keys"属性中创建VK，而不在"Key"节点上创建VK。

点击"Keys"属性下的<Browse> 按钮创建一个新的本地VK。

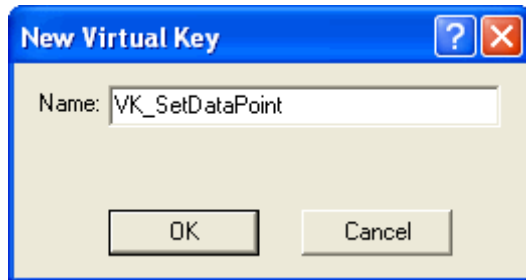


图. 37新的虚拟按键

此时，必须在控制中直接定义"Action"的属性。

当使用这种方法时，添加到"Keys"节点上的VK没有"Type"。这个VK可以用于其它页，但必须在控制中重新分配动作（本地VK）。

在图象编辑器的"Keys"表中，会显示全局及本地按键的按键分配。

Name ^	Type	Details	Layer
◆ VK_picGraphic		used globally	
◆ VK_picMainMenu		used globally	
◆ VK_picTextValue		used globally	
◆ VK_SetDataPoint	SetDatapoint	Value<none>. 0]	Default

Design | Layers | Keys | LEDs | Panel |

图. 38 图象编辑器 - Keys

任务：在页脚中将VK分配给按钮

将改变页面的三个按钮分配给相应的VKs。

编译项目并传送到目标系统。

结果

在目标系统中可以改变页面。



谁能识别第一个错误？

答案将在下一部分中说明。



## 6.7 静态和动态文本

在这一部分，我们将了解一下文本的类型及如何设计文本。

文本的类型：

- 静态文本  
在运行中不能改变的文本。
- 动态文本  
作为一个过程结果能改变的文本（一个控制变量的改变）。
- 列表框  
操作者从提供的列表中选出的文本。
- 字符串  
控制程序提供的文本（参看6.8数字和文字显示）。

### 6.7.1 静态文本

我们已经在页眉中创建了一个静态文本“Page description”，因此用户在使用另外的静态文本时没有任何问题。

现在必须使用动态文本，因为在三个页面上都已显示了静态文本。

### 6.7.2 动态文本

动态文本根据一个值来显示（文本）。

将静态文本转变为动态文本需要哪些步骤？

- 在"Page description"文本组中需要有多个文本。
- 需要有一个控制变量来代表页码。
- 将"TextSource"属性设置为"Multiple text"，而不是"Single text"。
- 使用"IndexDataPoint"属性连接页码变量。

## 可视化组件的实例项目

任务：创建一个控制变量"PicNumber"

在"panel"任务中创建一个类型为UINT的新控制变量"PicNumber"，并将它添加到VC编辑器的数据源中。



任务：全局项目设置——链接当前页码

当变量传送给VC编辑器后，变量就会连接到全局项目属性"Runtime" - "Page" - "CurrentDatapoint"。

在运行中，页码会写进这个变量。



问题

在VC编辑器中的哪里定义页码“Page number”？

答案

每页有一个属性索引。索引在运行时用来识别页面（例通过变量来改变页面）。

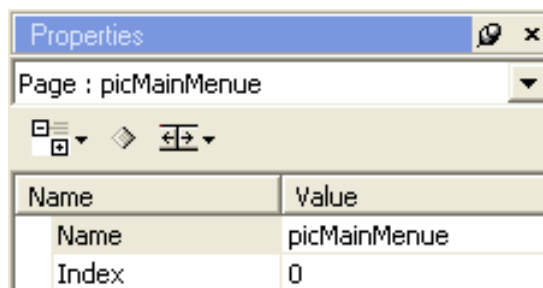


图. 39 页面属性- 索引

任务：将变量连接到“header”层中的文本控制

任务以这里显示的图象来执行。



在"PageDescription"文本组里缺少文本，是因为我们在前面只定义了"MainMenu"的文本。

Value	
Source	MultipleTexts
TextGroup	Countries
TextIndexOffset	0 : Germany
IndexDatapoint	Local.GLOBAL.ListBoxIndex

注意：

页面一旦创建就会分配页面索引。因此，索引应与页面描述相匹配。

任务：创建页面描述的文本

在"Page description"文本组中定义下面的文本：

Index ▲	English
0	Main Menue
1	Display Text and Value
2	Display Graphic Functions

图. 40 文本组– 页面描述

将项目传送到目标系统。

效果

正确的页描述会显示在每一页上。

## 6.7.3 列表框

列表框中会显示多个文本。这些文本是在控制任务中的文本组“Text group”或字符串变量区域“Field of String variables”中提供的。

列表框也用来从中选择文本，通过选择区域的每个文本来控制可见性，并且通过用户定义的输入动作向列表框中执行输入动作。

任务：创建一个文本组，最多带有10个按键

创建一个新的文本组（例：国家名）最多带有10个文本。在下一个练习中，文本组中的文本会显示在列表框中。



### 结果

Index ▲	English	German
0	Germany	
1	Austria	
2	United Kingdom	
3	France	
4	Spain	
5	Italy	
6	Portugal	
7	Sweden	
8	Netherlands	
9	Switzerland	

图. 41 文本组 – 国家

任务：为列表框创建一个控制变量

在用户任务“panel”中，需要一个新的控制变量“ListboxIndex”（UINT数据类型）来估计列表框的索引。

这个变量必须添加到可视化组件的数据源中。

任务：传送项目并测试列表框



结果

如果在运行时选中列表框，那么输入触摸板 "NavigationPad\_ver"就打开了。触摸板包括的按键有：

- 向上按键
- 向下按键
- 接受光标处的选项
- 取消按键

列表框的输入和输出行为

默认情况下，列表框作为"Input"列表框（即 "Input" 属性的值为"True"）。使用触摸板来选择文本。

如果"Input"属性设置为"False"，则只能用分配的变量"ListboxIndex"来选择文本。

## 6.7 数字和字母的显示

在这个例子中，我们只使用数字值来显示和输入。控制任务中的字符串或数组变量也可以输入和显示出来。培训模块TM650中会介绍字母的显示。

我们已经在主菜单中已创建了一个数字输入输出区域。

任务：在输出区域中显示变量"ListboxIndex"



在"TextValues"页面中插入一个数字输出区域并与变量"ListboxIndex"连接。

结果

在输出区域中会显示选中的文本索引。

## 6.8.1 数字输入的焦点行为

如果在一个页面上显示多个输入区域，那么对选中的输入区域（焦点）加亮是很重要的（尤其是通过键盘输入时）。

正如我们在"Main menu"页面中看到的，在"Input set value"属性中使用"Focus color"显示，而不用背景颜色显示。

在页面属性中，将焦点颜色设置为全局类型并应用于整个页面。

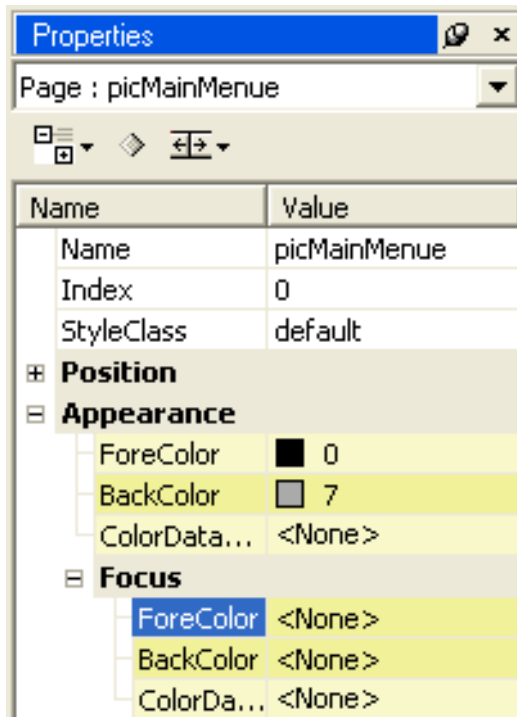


Fig. 42 属性 – Focus color

任务:用三个新的输入区域测试焦点行为



在"panel"任务中创建三个INT类型的变量"SetTemperature1", "SetTemperature2", "SetTemperature3".

在"TextValues"页面中创建三个输入区域并与三个新变量连接。

将这些改变传送到目标系统中并测试其行为。

结果

当完成一个输入时, 下一个输入区域就会成为焦点。

现在页面应如下所示:

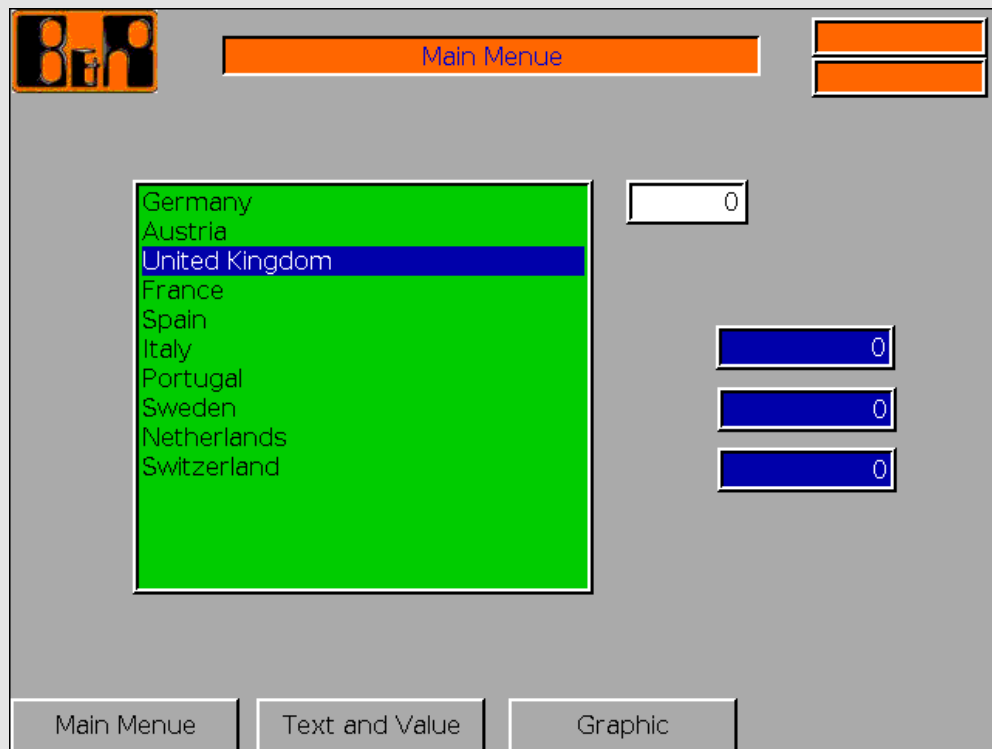


图. 43 屏幕页面—文本和数值

# 可视化组件的实例项目

## 问题

在使用触摸面板时，如果用户不希望出现焦点该怎么办？

## 答案

在输入区域和输入区域的风格（只要页面上的所有输入区域有相同的风格）中，将焦点颜色的前景和背景颜色设为一致。

如果不是这种情况，则把焦点的"Foreground color" 和 "Background color"设置为<None>。

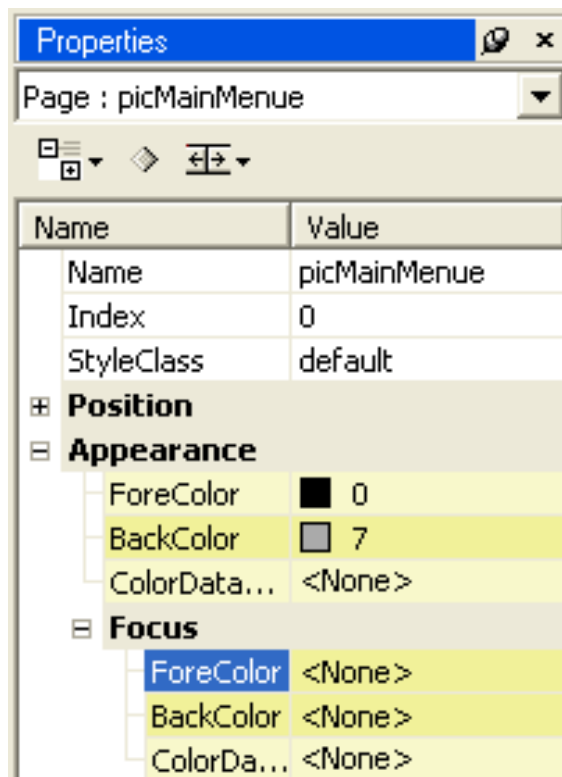


图. 44 全局属性- Focus



## 6.8 棒状图及其它图形对象

图形对象是图文显示中一种很重要的设计特征。在可视组件中可使用以下图形对象：

- 静态位图  
标志，背景图象，机械图表
- 动态位图  
在同一位置显示尺寸相同的不同位图。通过变量选择位图。
- 线条
- 外形  
矩形，圆或圆环
- 棒状图

这个实例已经加入一个"Logo"，（即"静态位图"）。不用单独学习线条和矩形，每个人都可以在"图形"页面中使用这些对象。

## 6.9.1 棒状图显示

棒状图有以下功能：

- 0 – 100 % 的过程指示
- 在棒状图图表中显示数值
- 用颜色显示数值的范围

棒状图最重要的属性：

- 可以显示变量的数值
- 排成直线
- 颜色设计
- 数值范围
- 起始值从0% 到 100%

练习：以棒状图表来显示数值

在前面的任务中，已创建了三个变量的输入区。现在要在 "Graphics"页面上将三个变量连接到三个"Bar Controls"。



这个棒状图将显示的数值范围应为0 到 100。棒状图的0点在底部（即 "Orientation"属性被设置为"Up"）。

棒状图不是分段显示时– "Segments"属性=0

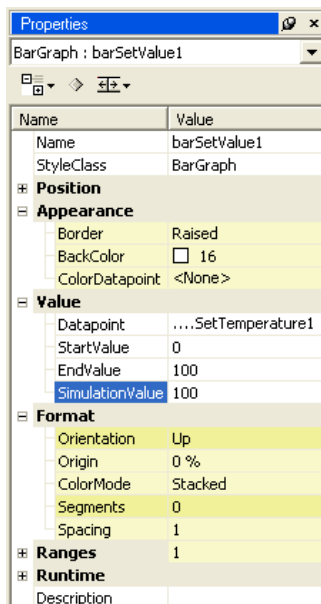


Fig. 45 棒状图属性

任务:添加三个改变数值的输入区

在"Graphics"页面上,从"Text and Value"页面上插入三个输入区域。



结果

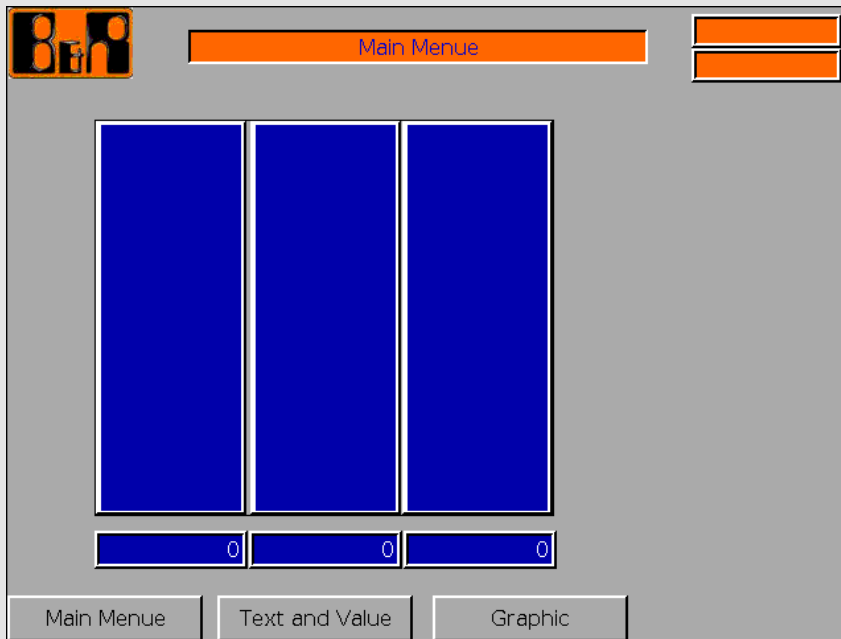


图. 46 截屏 - Graphics

任务:用棒状图表示正值和负值

现在已经可以显示0 到100之间变化的数值。如果向上和向下延伸滚动条,那么可以显示-100 到 +100之间变化的数值。

方法

改变"StartValue" 和 "Origin"的属性。

在顺利完成所有的练习和任务之后,针对这一培训模块的练习就结束了。

## 7、总结

完成这个培训模块之后，就可以使用可视组件编辑器来创建可视化应用了。

在这个可视化组件的基础培训模块中，并不能涵盖编辑器及其组件的所有功能和特征，因为可视化组件编辑器可以提供无数的可能性。用一个实例项目来解释编辑器的最重要的特征和元素，同时也提供了足够的信息来理解可视化组件的使用。

由于每个可视化应用都有不同的要求，所以可以在VC帮助中查找更多的信息。

想要全面了解可视化组件的知识，需要花些时间学习可视化组件的其它培训模块TM6xx。

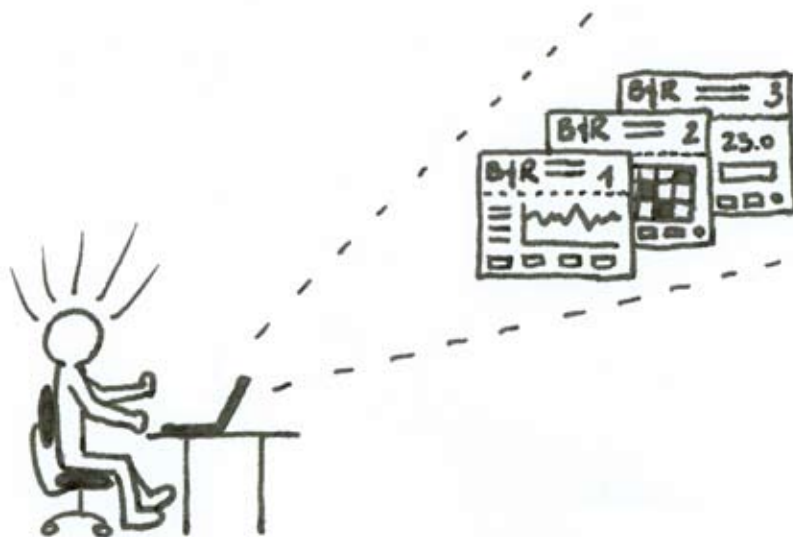


Fig. 47 可视化组件编辑器

Notes

## 培训模块综述

- TM200 – 贝加莱B&R 公司介绍\*\*
- TM201 – 贝加莱B&R 产品系列\*\*
- TM210 – Automation Studio™ 基础
- TM211 – Automation Studio™ 在线通信
- TM212 – 自动化对象 (Target) \*\*
- TM213 – 自动化运行 (Runtime) 系统
- TM220 – 维护信息\*
- TM221 – 自动化组件和出错信息查询\*
- TM223 – Automation Studio™ 诊断
- TM230 – 结构化软件编程
- TM231 – 面向机器设备的Automation Studio™ \*
- TM240 – 梯形图(LAD)
- TM241 – 功能块图 (FBD)\*
- TM242 – 连续功能图 (CFC)\*
- TM243 – 顺序功能图 (SFC)\*
- TM245 – 指令表 (IL)\*
- TM246 – 结构文本 (ST)
- TM247 – Automation Basic (AB)\*
- TM248 – ANSI C
- TM250 – 内存管理和数据存贮
- TM260 – Automation Studio™ 函数库I
- TM261 – Automation Studio™ 函数库 II\*
- TM264 – 定时处理单元 (TPU) \*
  
- TM400 – 运动控制的基础
- TM401 – 贝加莱B&R 运动控制产品\*\*
- TM402 – 运动控制系统的计算\*
- TM410 – ASiM 的基础
- TM440 – ASiM的基本功能
- TM441 – ASiM多轴运动功能
- TM445 – ACOPOS ACP10 软件
- TM446 – 电子凸轮\*
- TM447 – ACOPOS 智能过程技术 (SPT) \*
- TM450 – ACOPOS 控制理念和控制器设置
- TM460 – 启动B&R 电机\*
- TM461 – 启动第三方电机\*
- TM470 – CNC\*
  
- TM600 – 图文显示的基础
- TM601 – 贝加莱人机界面产品\*\*
- TM610 – ASiV 的基础
- TM620 – ASiV 的维护\*
- TM630 – 图文显示的编程规则
- TM640 – ASiV报警系统
- TM650 – ASiV的国际化操作
- TM660 – ASiV 的远程操作
- TM670 – ASiV 高级应用
  
- TM700 – Automation Net PVI
- TM701 – PVI 通信\*
- TM710 – PVI DLL 编程
- TM711 – PVI的服务
- TM712 – PVIControl.NET
- TM720 – PVI 维护和诊断\*
- TM730 – PVI OPC
  
- TM800 – APROL 系统概念
- TM801 – APROL 工程设计基础
- TM810 – APROL 安装, 配置和恢复\*
- TM811 – APROL运行(Runtime)系统\*
- TM812 – APROL 操作员管理
- TM813 – APROL XML 查询\*
- TM814 – APROL 审计追踪\*
- TM820 – APROL 维护\*
- TM830 – APROL 项目工程设计
- TM840 – APROL 参数管理和配方
- TM850 – APROL 控制器配置和INA 通讯
- TM860 – APROL 库设计
- TM861 – APROL 通讯互联\*
- TM865 – APROL 库指导手册
- TM870 – APROL Python编程\*
- TM880 – APROL 报表\*

\*\* 查看产品目录

\* 即将出版

## 全球总部

Bernecker+Rainer Industrie-Elektronik Ges.m.b.H.

B&R Straße 1

A-5142 Eggelsberg 奥地利

Tel.: +43(0)7748/6586-0

Fax: +43(0)7748/6586-26

info@br-automation.com

www.br-automation.com

## 中国总部

贝加莱工业自动化（上海）有限公司

上海市漕宝路70号光大会展中心C座16楼

Tel.: +86/(0)21/6432 6000

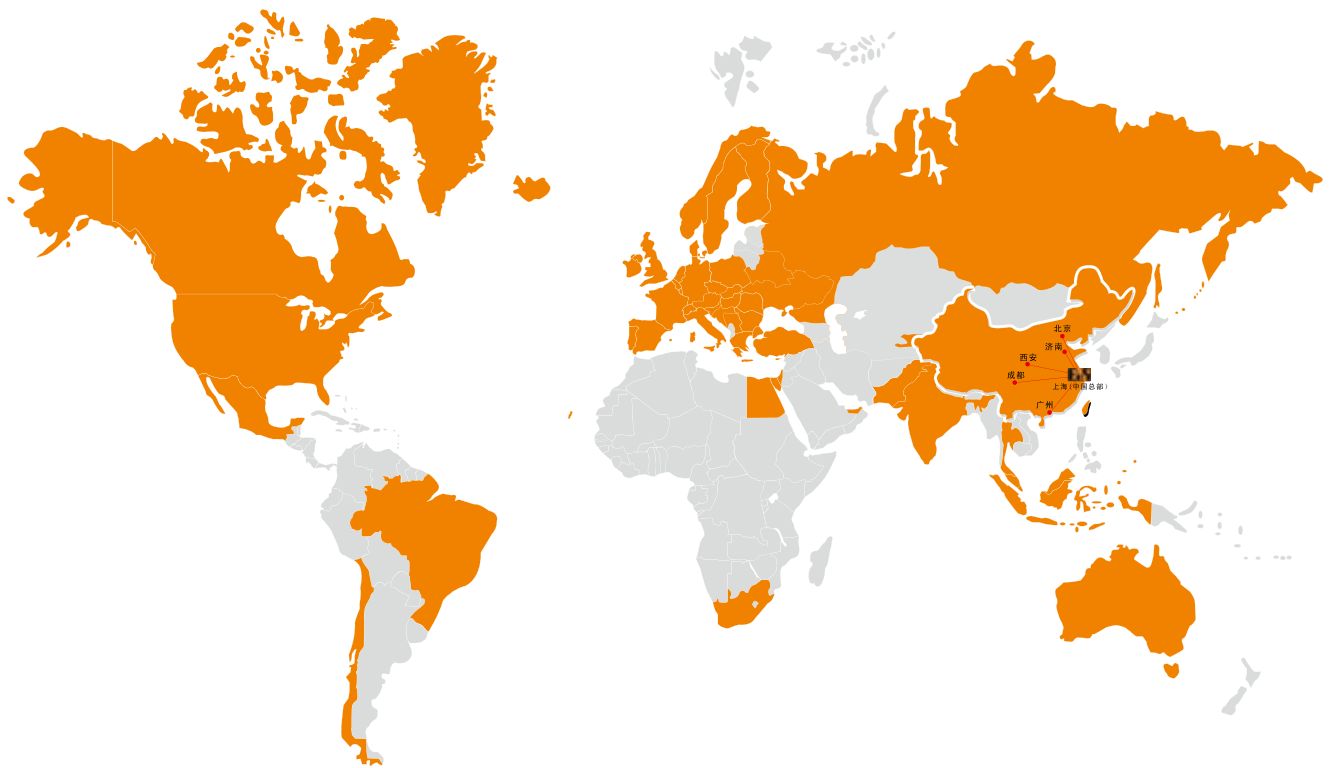
Fax: +86/(0)21/6432 6108

info.cn@br-automation.com

www.br-automation.cn

TM610FRE-25-CHN  
©2006 by B&R. All rights reserved.  
All trademarks presented are the property of their respective company.  
We reserve the right to make technical changes.

全球50多个国家超过120个分支机构 [www.br-automation.com/contact](http://www.br-automation.com/contact)



中国总部



中国办事处

Austria · Australia · Belgium · Belarus · Brazil · Bulgaria · Canada · Chile · China · Croatia · Cyprus · Czech Republic · Denmark · Egypt · Emirates · Finland · France · Germany · Greece · Hungary · India · Indonesia · Ireland · Israel · Italy · Korea · Kyrgyzstan · Malaysia · Mexico · The Netherlands · Norway · Pakistan · Poland · Portugal · Romania · Russia · Singapore · Slovakia · Slovenia · South Africa · Spain · Sweden · Switzerland · Thailand · Turkey · Ukraine · United Kingdom · USA